

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-288572
(43)Date of publication of application : 04.11.1997

(51)Int. Cl. G06F 9/06
G06F 9/46
G06F 12/00

(21)Application number : 09-035817 (71)Applicant : FUJITSU LTD
(22)Date of filing : 20.02.1997 (72)Inventor : YASUI SEIJI
OTA YOSHIO
MARUYAMA FUMIYUKI

(30)Priority

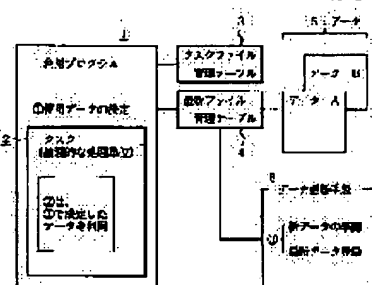
Priority number : 08 31584 Priority date : 20.02.1996 Priority country : JP

(54) DATA EXCHANGE DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To facilitate the exchange of data for every task of a use program without stopping the use program by registering data used at every task of the use program.

SOLUTION: The task 2 of the use program 1 refers to the latest file management table 4 in accordance with an access request and the latest flag detects data 5 of 'on' and registers information in a task file management table 3. Data 5 is accessed based on the task file management table 3 and a job processing is executed. In this case, a data update means 6 generates new data 5, registers it in accordance with the pertinent file name of the latest file management table 4. Then, the latest flag is turned on and the least flag of former data is turned off. Data which the task 2 started hereafter uses is dynamically exchanged.



BEST AVAILABLE COPY

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[The technical field to which invention belongs] This invention relates to data ON ***** which replaces the data which a use program uses.

[0002] The need of receiving continuous running of a computing system in recent years increases, the stop time of a use program is shortened as much as possible, and there is a demand which replaces the data which a use program uses. For this reason, the technique for replacing data is demanded, without completely stopping a use program.

[0003]

[Description of the Prior Art] After stopping a use program and changing old data to new data, it is made to reboot and is once made to prepare new data, and to perform operating processing, when replacing conventionally the data which a use program accesses and performs operating processing using new data. This is because there is a case where it becomes impossible for the logical conflict by having changed to occur and operate when the data which the use program is using working is replaced.

[0004]

[Problem(s) to be Solved by the Invention] In order to change old data to new data, to reboot and to perform operating processing conventionally after making a use program stop when replacing data as mentioned above, exchange of data and the reboot of a use program took time amount, and there was a problem that the demand of continuous running could not apply to a powerful system.

[0005]

[Means for Solving the Problem] With reference to drawing 1 , The means for solving a technical problem is explained. In drawing 1 , a task 2 is the example of a logical batch.

[0006] The task file managed table 3 is matched with a task, and information on a file which the task concerned uses is registered. The newest file control table 4 registers the newest file information in a system.

[0007] The renewal means 6 of data updates data in a file. Next, actuation is explained. Corresponding to an access request, the task 2 of a use program finds data of ON of the newest flag with reference to the newest file control table 4, registers that information into the task file managed table 3, accesses data based on this task file managed table 3, and is made to perform operating processing.

[0008] Under the present circumstances, while matching and registering with a file name to which the renewal means 6 of data creates new data, and the newest file control table 4 corresponds, he is trying to change dynamically data with which the task 2 which cleared ON and the newest flag of old data, and was started after this in them uses the newest flag to new data.

[0009]

[Embodiment of the Invention] Next, the gestalt of operation of this invention and actuation are explained to details one by one using drawing 4 from drawing 1 .

[0010] Drawing 1 shows system configuration drawing of this invention. In drawing 1 , the use program 1 is a program which performs various operating processings, generates two or more tasks 2 if needed (starting), and performs operating processing here.

[0011] A task 2 is a logical batch, and after generating a new task for every generating of tolan ZAKUSHO and completing the transaction concerned, it is made to terminate a task here.

[0012] The task file managed table 3 registers and manages the information on the file which matches with Task ID and the task of the task ID concerned uses (refer to (a) of drawing 2).

[0013] The newest file control table 4 is matched with a file name, registers the newest flag and data, and manages the newest file (the newest data). Here, that whose newest flag is ON is the newest data (the newest file) of a file name. It is old data (file) which has the off newest flag.

[0014] Data 5 is actual data stored in the file, and is Data A and Data B here. The renewal means 6 of data updates data, and registers the newest data into the newest file control table 4 (it mentions later using drawing 4).

[0015] Next, actuation of drawing 1 is explained. In drawing 1 , ** determines use data. This determines that the newest flag of the newest file control table 4 will use the data of the file name of "1" (ON). The determined contents are registered into the task file managed table 3. That is, the identifier of the newest data is registered corresponding to a file name.

[0016] ** Perform operating processing using the data (the newest flag is data (file) of "1" (ON)) (access) whose use program is a logical batch and which was determined by ** about the generated task. Thereby, a task 2 becomes possible [performing business using the newest data (the newest file) determined by **].

[0017] Next, it prepares, new data reaches and the renewal means 6 of data does the newest data registration of **. It sets the newest flag of old data as "0" (OFF) while matching this with the file name which had new data specified, registering it into the newest file control table 4 of (b) of drawing 2 mentioned later and it sets the newest flag to "1" (ON), and it changes data to the newest data dynamically. Since the newest flag after this registration performs business after this using the data of the file name of "1" (ON), the started task 2, the newly generated task 2 concerned it becomes possible to switch to the new data (file), to access the task 2, started henceforth, and to perform operating processing, without performing operating processing using the registered newest data, and stopping the use programs. It explains to details one by one below.

[0018] Drawing 2 shows the example of a table of this invention. (a) of drawing 2 shows the example of a task file managed table. This task file managed table 3 registers the newest data name corresponding to the file name which matches with Task ID and is used by the task concerned.

[0019] (b) of drawing 2 shows the example of the newest file control table. This newest file control table 4 is matched with a file name, and registers the newest flag and data. The newest flag expresses the purport which is the newest data at the time of "1" (ON), and expresses the purport which is old data at the time of "0" (off). Data is actual data and is data required in order for a task 2 to use it and to perform various operating processings.

[0020] Next, actuation of other examples is explained according to the sequence shown in the flow chart of drawing 3 . In one task

[0020] Next, actuation of other examples is explained according to the sequence shown in the flow chart of drawing 3. In one task (user's program), two or more transactions shall be processed sequentially, an OPEN instruction of a file shall be taken out with this example at the time of initiation of each transaction, and one transaction shall be completed through access processing of actual files, such as a READ instruction, and a CLOSE instruction of a file.

[0021] Drawing 3 shows the explanation flow chart of this invention of operation.

[0022] In drawing 3, as for S1, a use program issues an OPEN instruction of a file. By this, it will be opened by the file, for example, the file of a file name "X."

[0023] S2 acquires the task ID of a use program. Corresponding to generating of TORAZAKUJON new to a use program, a new task is generated and this is acquired from OS besides illustration of the task ID.

[0024] S3 registers the information on the newest file into the task file managed table 3 corresponding to Task ID. If the task file managed table 3 is already creation ending, the modification time of the task file managed table 3 will be compared with the modification time of the newest file control table 4, and if the renewal of the task file managed table 3 is unnecessary if the same, and the newest file control table 4 is newer, the task file managed table 3 will be updated.

[0025] S4 receives the notice of the access request from a use program.

[0026] S5 acquires the task ID of a use program. S6 performs access processing of data with reference to the task file managed table 3 based on Task ID.

[0027] It becomes possible to find the newest data for every task of a use program, and every transaction, to perform access processing about the newest data concerned, and to perform various operating processings by the above.

[0028] Drawing 4 shows the ON ** flow chart of the data of this invention. In drawing 4, S21 creates a new file (data). S22 registers a new file into this system.

[0029] S23 makes the newest the file which had registration directions among the newest file control tables 4, and makes an active file

ファイル名	最新フラグ	データ
X	1	A

old things. For this, the newest file control table 4 of (b) of drawing 2 is,

The newest data "B" is created, the newest flag of registration and newest data "B" is set to "1", the newest flag of old data "A" is changed into what was ***** from "1" "0", and it is performed as follows.

[0030]

ファイル名	最新フラグ	データ	最新フラグ	データ
X	0	A	1	B

After this, about the data "B" of "1", as for the generated task 2, this newest flag becomes possible [replacing data dynamically] so that it may use (access) and operating processing may be performed.

[0031]

[Effect of the Invention] Data can be dynamically changed easily to every [of a use program] task (logical batch), without stopping a use program like before, since the configuration which registers the data used every task 2 of a use program is adopted according to this invention as explained above. By these, even if a use program will need to replace data by the processing under operating employment, data is replaced easily dynamically, continuation employment can be carried out, the conventional use program can be stopped, data can be replaced, and a temporary systems operation halt by a series of processings of a reboot can be lost now.

[Translation done.]

CLAIMS

[Claim(s)]

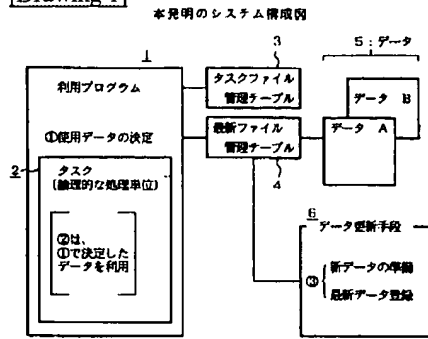
[Claim 1] Data ON ***** which is characterized by providing the following and which replaces data which a use program uses A run unit file control table which registers a logic name and a physical name of data for every logical run unit of a use program The newest file control table which registers a physical name which is a name of live data corresponding to a logic name of data which a use program uses An access-control means to access data which corresponds with reference to a run unit file control table which updates the contents of said run unit file control table based on said newest file control table, and corresponds in a logical break of a use program corresponding to an access request

[Claim 2] Data ON ***** according to claim 1 characterized by a logical break of said user's program being processing initiation of a new transaction.

[Translation done.]

DRAWINGS

[Drawing 1]



[Drawing 2]

本発明のテーブル例

(a) タスクファイル管理テーブル 3

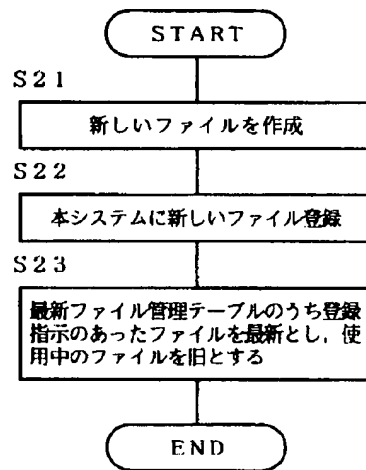
タスクID	使用しているファイル名	データ
01	X	A
02	Y	C

(b) 最新ファイル管理テーブル 4

ファイル名	最新フラグ	データ	最新フラグ	データ
X	1	A	0	B
Y	1	C	0	D

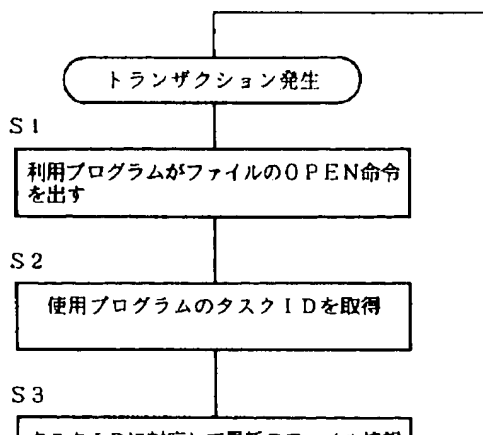
[Drawing 4]

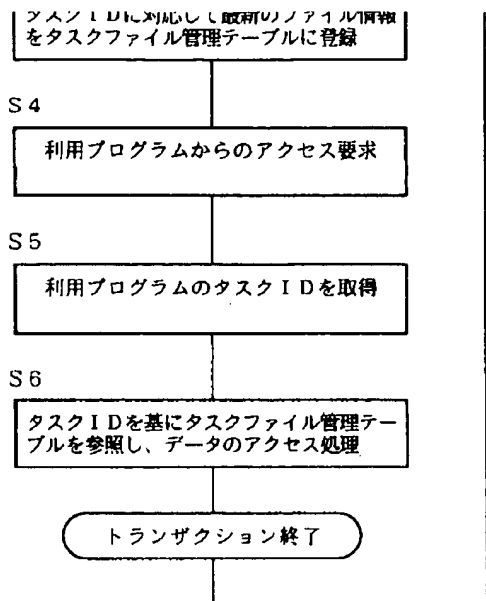
本発明のデータの入換フローチャート



[Drawing 3]

本発明の動作説明フローチャート





[Translation done.]

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☒ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.